The journal of Apple technology.

Volume Number: 17 (2001)
Issue Number: 4
Column Tag: Viewpoint

Viewpoint

By Rich Morin

In this month's Viewpoint, Rich Morin revisits some themes that he discussed in our first Viewpoint, two years ago. Fortunately, Apple has now released Mac OS X, so much of what used to be speculation is now reality.

I am an unapologetic fan of both Mac OS and Unix. My desktop system (cerberus) is a three-headed G3, running Mac OS. A FreeBSD server sits downstairs, providing support for email, ftp, telnet, etc. The Mac provides a pleasant user interface and a fine set of "productivity tools"; the server gives me a reliable platform for my Unix-based programming projects.

Although I find this to be a very workable combination, it can prove awkward on occasion. The FreeBSD and Mac OS environments are not tightly coupled, so moving information between them can be a nuisance. Also, Mac OS apps have a tendency to use undocumented, proprietary file formats. The idea of an integrated system, based on both Mac OS and Unix, is therefore quite appealing to me.

More to the point, I once had such a system. A/UX, which ran on the Mac II and follow-on machines (e.g., Quadras), was a very promising blend of Mac OS and Unix. Unfortunately, Apple never gave the product any real marketing support, so it never built a large customer base. Nonetheless, it was a useful and interesting "proof of concept", solving some hard problems in an elegant fashion and offering some real innovations.

A decade later, hardware advances have given us Macs that are a thousand times faster than a Mac II, disk drives that are a thousand times larger, etc. This computing power has allowed a number of "hideously inefficient" software innovations (e.g., Interface Builder, Java, OpenGL, XML) to become practical. Bringing all of this together, Apple is set to try again, in a much more splendid fashion. Apple's engineers have managed to combine great engineering with a very stylish and appealing appearance. I think they are poised to shake up the industry.

Nonetheless, Apple faces some real problems in presenting this mixture to the world. They want it to be accepted as "the next generation of Mac OS" by their users and developers. If they allow the Unix infrastructure to show through, they may scare away their traditional user base. Even talking up the Unixish benefits of the product (e.g., stability, performance, industry standard interfaces) may work against the aim of easy acceptance by their current users and developers.

On the other hand, these same Unixish aspects are critical to bringing in users and developers from Linux, Solaris, etc. Two years ago, I dared to predict that "Apple will cater to, and actively woo academic, research, and otherwise technical users; in short, the bastions of the Unix community. Apple may be quiet about it, but mark my words; it will be a large part of their planning." Apple has, in fact, been so quiet about Unix that it might as well have disappeared from their plans, entirely. They also made some discouraging decisions, such as killing off any possibility of case-sensitivity from HFS+.

On the other hand, a few positive indications have also surfaced. For one thing, Apple opted to keep UFS around, providing case-sensitive file names for situations that really need them. Then, at WWDC, Steve Jobs said that Mac

OS X used a "Linux-like" kernel. At MacWorld, Apple actually used the "U-word" on the large banner that sat atop their pavilion. Finally, and most usefully, Apple made its developer tools available to Mac OS X Beta users, without demanding a stiff entry fee that might have discouraged casual developers.

None of these indications prepared me, however, for Apple's recent moves. Making the Unix command line (and a full suite of commands) available to all users is really a bold move for a company that preaches (and practices) extreme simplicity in user interfaces. Shipping a developer CD-ROM in the Mac OS X distribution is even more radical; what do they think the typical Mac OS user is going to do with Interface Builder?

Well, Joe and Sally Sikspak may not want to load the developer disc, but their son or daughter may. The BSD command set found in Mac OS X is almost indistinguishable from that of the Linux machines the kids have been using, so they'll find the basic tool set quite comfortable. In addition, however, they will be able to try out Interface Builder (a real step up from hand-coding X11 calls), Project Builder, and Apple's other innovations. In short, kids who have been hacking on Linux systems for years should find Mac OS X to be a real treat!

If Joe or Sally happens to be an engineer, the kids may have some competition for time on the machine. But then, Power Macs are cheap enough that each member of the family may have one. They are also cheap enough that Joe or Sally may decide to replace the Sun or SGI box on their desk with a G4 cube. Apple may claim that it isn't in the workstation business, that that won't keep people from buying Macs and using them as workstations!

In any case, it appears that Apple is finally ready to acknowledge the value that the Unix user and development community (formal and informal) has to offer. By making powerful development tools available to all users, they hope to entice all sorts of Unix aficionados to consider Mac OS X as a real platform. If they succeed in their effort, the results will bring some real changes to the Mac OS developer community.

Traditionally, Macintosh developers have constituted a vanishingly small percentage of the overall community. Let's say that there are 24 million Apple users. MacTech, the technical publication for developers, has a circulation of something like 20,000 - less than of the entire user community. These are generally the more serious developers in the community. Even if we expand the definition of "developer" to be as broad as possible, we only get a total of 150,000 developers (still less than 1% of the Mac community).

Now, let's look at the Unix community (including Linux, *BSD, Solaris, etc). I don't know the total size of this community (nobody does, actually :-), but it's clearly in the millions and quite possibly in the tens of millions. Now, how many of these folks are "developers"? Well, that's a hard question, because the definition is a lot fuzzier than is is in the Mac community.

I would guess that at least half of the Unix community (i.e., users, system administrators, programmers) have written some shell or Perl scripts. I would not be surprised to learn that the majority of Unix users have written and compiled a few C programs. Just using the command line, which essentially all Unix users do, is in fact closer to programming than most Mac users ever come. In short, it's a very different user community.

Capturing a million or so of these users for the Mac could be a real coup for Apple. The raw numbers aren't that impressive; a million additional users would only add about 4% to Apple's user base. Most of these folks would qualify, however, as potential Mac OS X developers. They might be scientists, engineers, students, or hobbyists, but many of them are in a position to create new software (and retrofit old software) for the platform.

Although existing Mac OS developers could view this as unwanted competition, I think this would be misguided. Instead, I hope that the current band of developers will recognize the newcomers as a source of cross-fertilization and useful lore. Unix is, after all, an enormous, decades-old, distributed laboratory for computer science research. It would be very surprising if the old-line Unix hackers didn't have some interesting tricks to show the Mac OS developers.

Also, the Mac community has been largely unaffected by the "Free Software" and "Open Source" movements.

Freeware, in Mac parlance, is proprietary, binary-only software which is distributed at no charge. This kind of distribution is almost unknown in Unix circles, where "free software" always allows inspection, modification, and redistribution of the source code. An influx of Unix-style free software might well change the thinking of many Mac OS users and developers.

By the same token, Mac OS developers have a great deal to show (and tell) the newbies. Mac OS X may change some things about the UI, but a lot of the principles of Mac interface design will carry right through into the new system. More to the point, if we want Aqua to avoid the anarchic state of X11 applications, some advice and counsel will be quite necessary!

In short, it's a win-win situation. We all get to play with some really nifty development tools, on some really terrific hardware. We also get to try out thousands (literally!) of Open Source programs, either under their original command-line and/or X11 user interfaces or with new, Aqua-based front ends.